

In the claims:

1 (Currently Amended) A multi-thread processor for processing a plurality of threads comprising:

5 a Thread ID generator producing a unique thread indication for each said thread;

a plurality of register sets, one said register set for each said thread, each said register set for each said thread comprising a plurality of registers;

10 an n-way register set controller coupled to said plurality of register sets and simultaneously handling multiple read or write requests for one of more of said unique threads;

a Fetch Address Stage for the generation of Program  
15 Memory Addresses;

a Program Access Stage for receiving Program Memory Data associated with said Program Memory Addresses;

a Decode Stage for converting said Program Memory Data into instructions, said Decode Stage coupled to said n-way  
20 register set controller;

a First Execution Stage for handling a multiply class of instruction received from said Decode Stage;

A Second Execution Stage for handling an Arithmetic Logical Unit class of instructions received from said Decode  
25 Stage, said Second Execution Stage also coupled to said n-way register set controller;

A Memory Access Stage for handling reading and writing of external memory;

A Write Back Stage coupled to said n-way register set controller for writing data to said register set;

5 each said Stage performing an operation during a Stage Cycle;

said n-way register controller allowing simultaneous access to said plurality of register sets by at least two of said Decode Stage, said First Execution Stage, and said  
10 Write Back Stage;

said Thread ID value alternating from one stage cycle to the next.

2 (Original) The processor of claim 1 where a pipeline  
15 core is formed by stages in succession: said Fetch Address stage, said Program Access stage, said Decode stage said First Execution stage, said Second Execution stage, said Memory Access stage, and said Write Back stage.

20 3 (Original) The processor of claim 1 where said n-way register set controller simultaneously receives at least one of read requests from said Decode stage, read and write requests from said Second Execution stage, or write requests from said Write Back stage.

25

4 (Original) The processor of claim 1 where said Memory  
Access stage is coupled to a memory controller.

5 (Original) The processor of claim 4 where said memory  
5 controller issues a stall signal when receiving a memory  
request to an external memory.

6 (Original) The processor of claim 4 where said memory  
controller issues a stall signal when receiving a memory  
10 read request to an external memory.

7 (Original) The processor of claim 4 where said memory  
controller issues a stall signal which lasts an interval  
from receiving a memory read request to receiving requested  
15 data form said external memory.

8 (Currently amended) The processor of claim 1 where  
two threads are concurrently processed and said pipeline  
core comprises a subset of said stages operative on one said  
20 thread and remaining said stages operative on said other  
thread.

9 (Original) The processor of claim 1 where said first  
execution stage performs multiply operations and said second  
25 execution stage performs non-multiply instructions.

10 (Original) The processor of claim 1 where said  
decode stage forwards non-multiply operands to said second  
execution stage.

5 11 (Currently amended) The processor of claim 1 where  
program memory contains a single instance of a program, and  
said thread ID can be read by said program.

12 (Original) The processor of claim 1 where said  
10 thread ID can be read by each said thread.

13 (Original) The processor of claim 1 where each said  
thread reads said thread ID to perform thread operations  
which are independent.

15

14 (Currently amended) The processor of claim 1 where  
thread ID is used along with an address to enable ~~decode~~ a  
device in a memory map.

20 15 (Currently amended) The processor of claim 1 where  
devices are ~~decoded~~ enabled in a memory map based on address  
only.

16 (Original) The processor of claim 1 where said  
25 Decode stage performs decoding of instructions for said  
multiply class of instruction, and said First Execution

stage performs decoding of instructions for said arithmetic logical unit class of instructions.

17 (Original) The processor of claim 16 where if one of  
5 said multiply class of instructions requires a register operand, said operand is provided from said registers to said decode stage.

18 (Original) The processor of claim 16 where if one of  
10 said arithmetic logical unit class of instructions requires a register operand, said operand is provided from said registers to said first execution stage.

19 (Original) The processor of claim 1 where at least  
15 one said stage includes an operational clock which is at a higher rate than said stage clock.

20 (Currently amended) A multi-threaded processor comprising a plurality of stages operating on a stage clock  
20 for passing information from stage to stage, each stage including inter-stage storage for thread information associated with a thread ID;

a first said stage receiving program counter address information from a unique program counter address  
25 information from a unique program counter for each said thread ID and delivering said address to a program memory;

a second stage for receiving program data from a program memory.

a third stage for performing decode of said program data;

5 a fourth stage for performing multiplication operations or decode operations;

a fifth stage for performing non-multiplication operations;

a sixth stage for accessing external memory;

10 a seventh stage for writing results of computations performed in said fourth stage or said fifth stage back to a register set;

said register set being duplicated for each said thread ID;

15 said register set allowing simultaneous access by at least two of said third stage, said fourth stage, and said seventh stage;

each said first through seventh stage receiving said thread ID and operating according to a first or second  
20 value.

21 (Original) The multi-threaded processor of claim 20 where said first, third, fifth and seventh stages use one value for said thread ID, and said second, fourth, and sixth  
25 stages use a different value for said thread ID.

22 (Original) The multi-threaded processor of claim 20  
where said threads each control execution of a program, and  
said programs execute independently of each other.

5 23 (Original) The multi-threaded processor of claim 22  
where one solid thread may stop execution and the other said  
thread continues execution.

24 (Original) The multi-threaded processor of claim 20  
10 where said registers and said stages contain data which is  
used separately for each said thread ID.

25 (Original) The multi-threaded processor of claim 20  
where said stages alternate between two threads on each said  
15 stage clock.

26 (Original) The multi-threaded processor of claim 20  
where said thread-ID identifies a register set and a program  
counter.

20

27 (Original) The multi-threaded processor of claim 20  
where said third stage performs said decode for multiply  
operations.

28 (Original) The multi-threaded processor of claim 20  
where said fourth stage performs said decode for non-  
multiply operations.

5 29 (Currently amended) The multi-threaded processor of  
claim 27 28 where said fourth stage performs said multiply  
operations.

30 (Original) The multi-thread processor of claim 28  
10 where said fifth stage performs said non-multiply  
operations.

31 (Original) The multi- thread processor of claim 28  
where said non-multiply operations include at least one of  
15 rotate, shift, add, subtract, or load.

32 (Original) The multi-thread processor of claim 29  
where said multiply operations include multiplication by a  
constant from one of said registers.

20

33 (Original) The multi-thread processor of claim 30  
where said non-multiply operations include addition of a  
multiply result from said fourth stage.

25



34 (Original) The multi-thread processor of claim 20  
where said thread ID includes a plurality of values, each  
said value having at least one register and a program  
counter.

5

35 (Original) The multi-thread processor of claim 20  
where said sixth stage said external memory responds in more  
than one said stage clock cycle.

10

36 (Original) The multi-thread processor of claim 20  
where said external memory generates a stall signal for each  
said thread ID, thereby causing all said stages to store and  
maintain data for that thread ID until said stall signal is  
removed by said external memory.

15

37 (Original) The multi-thread processor of claim 20  
where said fifth stage generates an address for a data  
memory.

20

38 (Original) The multi-thread processor of claim 37  
where said sixth stage receives and generates data for said  
data memory.

25

39 (Original) The multi-thread processor of claim 20  
where said thread information storage includes registers

which store results from said fifth stage for each said thread ID.

40 (Original) The multi-thread processor of claim 20  
5 where said registers which store results from said fifth stage allow a non-stalled thread to continue execution without modifying said stored results.